

GNG2101

Design Project User and Product Manual

SimplyText User Manual

Submitted by:

B22 - SimplyText

Danny Harani, 300227739

Jeremy Dufour, 300231265

Tsz Ho (Gordon) Man, 300172813

Layan Omar, 300249213

11/12/2022

University of Ottawa

Table of Contents

List of Figures	III
List of Tables	III
List of Acronyms	III
1 Introduction	1
2 Overview	1
3 Getting Started	2
3.1 Configuration Considerations	3
3.2 Accessing the System	3
3.3 Exiting the System	3
4 Using the System	3
4.1 Document Capture	3
4.1.1 Saving Extracted Text	4
4.1.2 Flipping Through Cameras	5
4.1.3 Clearing The Screen	5
4.2 Search	5
4.2.1 Searching Extracted Text	5
4.2.2 Using Search Filters	6
5 Troubleshooting & Support	6
5.1 Unexpected Behaviors	7
5.1.1 No Camera Preview	7
5.1.2 No Search Result	7
5.2 Support	7
6 Product Documentation	8
6.1 Authentication	8
6.1.1 Bill of Materials	8
6.1.2 Equipment list	8
6.1.3 Instruction	8
6.2 Optical Character Recognition (OCR)	9
6.2.1 Bill of Materials	9
6.2.2 Equipment list	9
6.2.3 Instruction	9

6.3 Search and Filters	9
6.3.1 Bill of Materials	10
6.3.2 Equipment list	10
6.3.3 Instruction	10
6.4 Capturing and Saving	10
6.4.1 Bill of Materials	10
6.4.2 Equipment list	10
6.4.3 Instruction	10
6.5 Testing and Validation	11
6.5.1 Cloud Vision Tests	11
6.5.2 Tesseract Tests	12
6.5.3 Search Tests	12
6.5.4 Authentication Tests	12
7 Conclusion and Future Work Recommendations	13
7.1 Multi-Group Support	13
7.2 Improving OCR Accuracy	13
7.3 Highlighting Detection	13
7.4 Table Detection	13
7.5 Conclusion	14
8 Appendix I: Design Files	15

List of Figures

Figure 1. Final Prototype Layout	1
Figure 2: Image of SimplyText Login Screen	2
Figure 3. Camera Access Request	3
Figure 4: Image of SimplyText Capturing Screen	4
Figure 5: Captured Image and Extracted Text	4
Figure 6: Image of SimplyText Search Screen	5
Figure 7: Inspect View of Search Result	6
Figure 8: Modifying Search Filters	6

List of Tables

Table 1. Support Personnel	7-8
Table 2. Bill of Materials for Authentication	8
Table 3. Bill of Materials for OCR	9
Table 4. Bill of Materials for Search and Filter	10
Table 5. Bill of Materials for Capturing and Saving	10
Table 6. Cloud Vision Test Results	11
Table 7. Search Tests	12
Table 8. Design Documents	15
Table 9: Hyperlinks	19

List of Acronyms

Acronym	Definition
BOM	Bill of Materials
MAU	Monthly Active Users
OCR	Optical Character Recognition
ST	SimplyText

1 Introduction

This User and Product Manual (UPM) provides the information necessary for control panel operators to effectively use SimplyText and for prototype documentation. The user manual consists of a list of instructions and a detailed breakdown of the prototypes and how it was put together. The document includes a complete list of materials used, how the prototypes were created and evaluated along with the feedback provided by the users. The document explains: Considerations, how to access the system, organization and navigation, how to exit the system, various functions and features, sub-functions and sub-features, error messages that are displayed and what they mean, special considerations, maintenance, and support. The 4 subsystems have a detailed bill of materials, equipment list and instructions specific to each subsystem and a complete list of all design files.

2 Overview

User problem is that there is a need for an OCR scanning software to store medical data safely and securely. While also retaining a high accuracy of text recognition and the ability to scan and define different symbols. With that, the problem statement will be: The client has a large amount of medical data related to his son that must be converted to a digital format which allows for further analysis and easier searching. This will be done via a mobile application used by care workers.

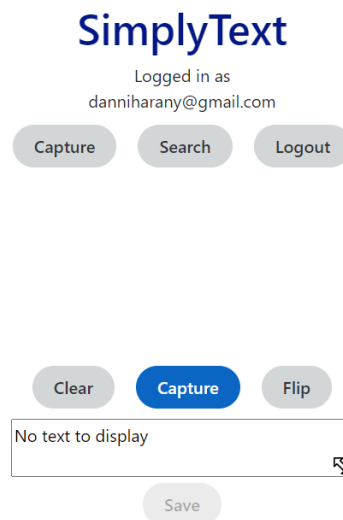


Figure 1. Final Prototype Layout

The fundamental need of the user is a scanning app that is easy to use. Furthermore, other client's needs can be categorized into three aspects: functionality, useability and form.

For functionality, the OCR has to have an accuracy that is over 90% and can be trained to improve results overtime, whilst having the data being easily searchable yet secure at all times.

For useability, the software needs to be simple to control and use, yet powerful at the same time for advanced users.

In terms of form, the client would like the product to be vastly available on most platforms, such as android and ios, an easy process to train the software, and has the ability to quickly recognize new users.

3 Getting Started

To get the software started, first open the software in the computer. Where a login page will look like something below:

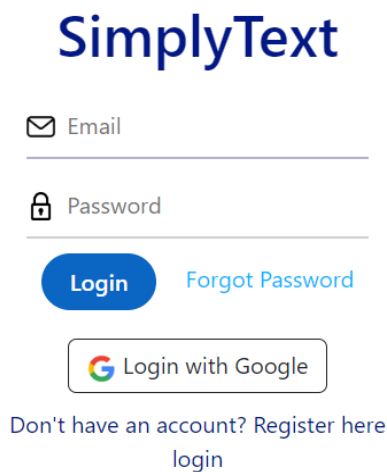


Figure 2: Image of SimplyText Login Screen

1. Username input

This area lets you input your username (ie. your email address) to access your account.

2. Password input

This area lets you input your password to access your account.

3. Login Button

This button completes the procedure by letting you login to access your account. After pressing it, the system detects if the credentials are correct to ensure the correct user access.

4. Register Link

This button redirects the user to another page to create an account if they do not have one.

3.1 Configuration Considerations

It is critical for SimplyText that your device has a camera. When accessing the Capture screen of the website, make sure to allow access to the camera as seen below.

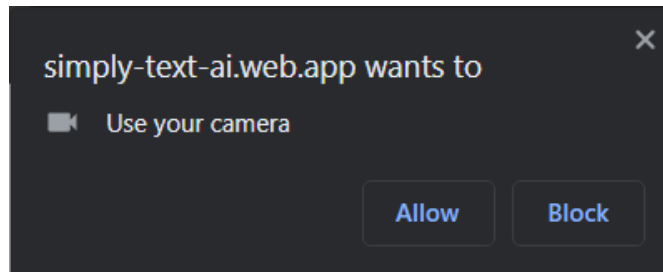


Figure 3. Camera Access Request

Press allow to enable capture functionality. Other features are not blocked if you do not wish to allow camera access.

3.2 Accessing the System

Before logging into the system, a user must ensure they have registered. This can be done by pressing on “Register Here” on the login page. Users will be promoted to enter a name, email and password. A user will be automatically logged in when they register successfully. If a user forgot their access password, they can enter a registered email into the forgot password field visible after pressing “Forgot Password” on the login screen.

3.3 Exiting the System

A logout button at the top of any screen will allow a user to exit the system. Note that closing the browser window does not logout a connected user.

4 Using the System

SimplyText has two (2) main features: Document Capture and Search. The following subsections provide detailed, step-by-step instructions on how to use the various functions and features of the SimplyText application.

4.1 Document Capture

Capturing a document is done on the Capture screen. A camera preview will be shown on the screen. An example view is shown below.

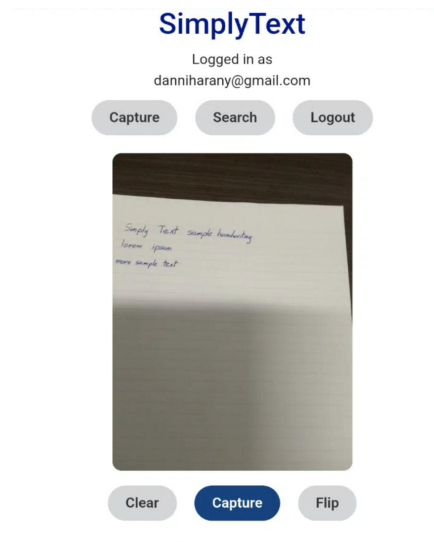


Figure 4: Image of SimplyText Capturing Screen

Here, there are three (3) options: Capture, Clear and Flip. Selecting Capture below the camera preview will take a screenshot of said preview and display it under the button row. Text found within this screenshot will be extracted and displayed in a modifiable text field as seen in figure 4.



Figure 5: Captured Image and Extracted Text

4.1.1 Saving Extracted Text

Once an image was captured and text was successfully extracted, the save button will become enabled. Before saving, a user should ensure the validity of the extracted text and make any modifications necessary. Pressing save will upload the image, originally extracted text and user-edited

text to a secure online database. A popup will inform the user of the status of the save (i.e. success/failure),

4.1.2 Flipping Through Cameras

The Flip button will attempt to alternate between environment facing and user facing cameras (Environment facing camera is selected by default). A blank preview is displayed if a requested camera is not found or the appropriate permissions are not provided in the browser.

4.1.3 Clearing The Screen

Pressing the Clear button in Figure 3 will clear the extracted text field and delete the screenshot, essentially clearing all data provided from capturing the photo. If a screenshot and its data were saved, clearing will not delete the saved data.

4.2 Search

The search feature allows a user to access previously saved documents. A user can access documents saved by anyone on their team.

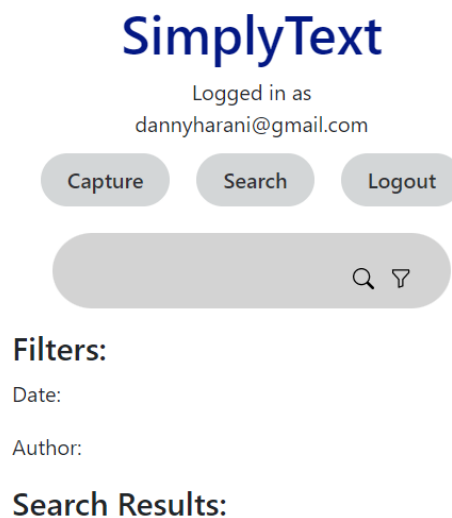


Figure 6: Image of SimplyText Search Screen

4.2.1 Searching Extracted Text

To access the extracted text files, the keywords wanted to filter from the saved text are typed onto the search bar. A user must press on the magnifying glass to initiate the search. Documents matching the query will display under the search results. For a more detailed view of a result, a user can press on it and the complete extracted text and matching image will be displayed as seen below.

Inspect:
 Author: danniharany@gmail.com
 Date: 2022/12/1
 Content:
 Simply Text
 Sample text can be recognized
 With okay accuracy.
 The accuracy will improve over time with usage
 Dec/01/2022

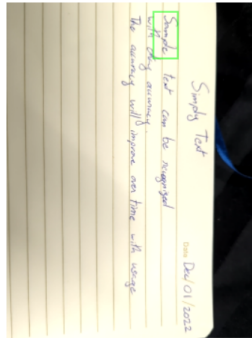


Figure 7: Inspect View of Search Result

4.2.2 Using Search Filters

The queries can be also done by entering the date or author of the scans being done. Pressing on the funnel icon in the search field will display the Author/Date filter options. To click in, press the date/author on the screen and enter the relevant words or values to initiate the search. Documents matching the query will display under the search results.

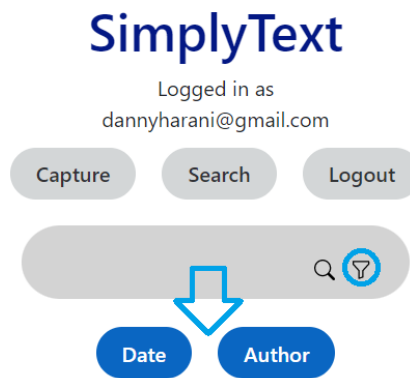


Figure 8: Modifying Search Filters

5 Troubleshooting & Support

This section will provide troubleshooting information and support methods for the SimplyText application.

5.1 Unexpected Behaviors

5.1.1 No Camera Preview

If no camera preview is visible on the Capture screen of the application, ensure to follow these steps:

1. Ensure your device has an on-device camera (not tested for external webcams)

If your device has a camera, go to the next step. If not, the capturing functionality is not compatible with your device

2. Ensure the browser has access to the camera

In your system's settings, ensure the browser is allowed access to the camera. Visit your system's help menu for information on how to enable camera access for your browser. SimplyText should now prompt you for camera access.

3. Ensure the browser has allowed SimplyText the required camera access

In most browsers, this can be done in *Settings > Privacy & Security > Permissions*, or by pressing on the lock icon in the url bar and pressing *Site Settings*. Ensure SimplyText is permitted access to the camera. If this is your first time accessing the Capture Screen, a popup should automatically show up, press allow.

4. The image is still not displaying.

If a preview is still not displaying try flipping the camera by pressing on Flip as seen in Figure 4. Contact support if the camera preview is still not displaying.

5.1.2 No Search Result

If no search result is displayed when you select the search (magnifying glass) icon, try broadening your search. Ensure proper capitalization, SimplyText search is case-sensitive. If a date filter is used, ensure use of this date format: YYYY/MM/DD. Omit leading zeros, for example December 6th 2022 is entered like this 2022/12/6. Ensure Author filter is the email associated with the user that uploaded the document, their name will not match any result.

5.2 Support

Table 1. Support Personnel

Help With	Name of Contact	Email
Administration	Gordon	tman018@uottawa.ca
Administration	Layan	lomar018@uottawa.ca
Capturing and Saving	Danny	dhara044@uottawa.ca

Help With	Name of Contact	Email
Registration and Login	Danny	dhara044@uottawa.ca
Registration and Login	Jeremy	jdufo070@uottawa.ca
Searches and Filters	Jeremy	jdufo070@uottawa.ca

6 Product Documentation

6.1 Authentication

Firestore was our choice of Authentication Client. Not only is it a well known client, our product also uses other Firestore features. Other options were considered, such as auth0 by Okta but quickly discarded since they have no major benefit over Firestore. The implementation of Firestore authentication was done to keep a more concise technology stack.

6.1.1 Bill of Materials

Table 2. Bill of Materials for Authentication

Part	Description	Cost Type	Cost	Exception
Authentication Client	Allows for user authentication	Variable	0.0055\$/MAU	Free for 50 000 MAU

6.1.2 Equipment list

1. Firestore
2. Firestore NPM package

6.1.3 Instruction

To include Firestore authentication to a web application, this [documentation](#)¹ from Firestore was followed. In the case of SimplyText, slight deviations were made from the directions to allow compatibility with the React framework. Firestore has an npm package that facilitates integration. The [readme](#)² contains directions on its usage. After setting up firestore, follow these steps

1. Create a login and register page
2. Add required fields to these pages (i.e. email, password, confirm password)
3. Request authentication on button press (login/register buttons) and await Firestore response.
4. User is now authenticated.

6.2 Optical Character Recognition (OCR)

SimplyText uses two (2) services for its optical character recognition. Many were available on the market but most were expensive or did not have desired functions (table detection, trainability, etc). Google Cloud Vision is the first system to be used and is the only system to display output. Cloud Vision is very high accuracy for neat handwriting and printed text. However, it is not trainable. Tesseract OCR is another system. This OCR system runs on-device or can be deployed as an API. The default model is not accurate with handwriting, but Tesseract can be trained to learn new handwriting.

6.2.1 Bill of Materials

Table 3. Bill of Materials for OCR

Part	Cost Type	Cost	Exception
OCR Technology CloudVision	Variable	\$1.5/month/1000 units	Free for first 1000 units
OCR Technology Tesseract	Fixed	Free, Open source	N/A

6.2.2 Equipment list

1. Cloud Vision Extension on Google Cloud platform
2. Tesseract JS NPM package

6.2.3 Instruction

1. For Cloud Vision, a valid Google Cloud project is required. Instructions on how to set up a cloud project can be seen [here](#)³.
2. The Cloud Vision api has to be connected to the project setup, and billing must be enabled. For further instructions on billing and setting up Cloud Vision with a project, view this [documentation](#)⁴.
3. The Cloud Vision API call was done through the use of a Firebase Cloud function. Instructions on the implementation of a cloud function can be found [here](#)⁵.
4. An API call with a valid base-64 image is passed to the Cloud Function. More information on the call structure and return result can be found [here](#)⁶.

6.3 Search and Filters

Implementation of TypeSense was not completed for this version of SimplyText. Instead, a rigid linear search was implemented.

6.3.1 Bill of Materials

Table 4. Bill of Materials for Search and Filter

Part	Description	Cost Type	Cost	Exception
File Read	Reading from file database	Variable	\$0.033/100 000 reads/day	Free for first 50 000 reads

6.3.2 Equipment list

1. Firestore database

6.3.3 Instruction

1. Have a Firebase Firestore Collection setup in which to search.
2. Follow [these](#)⁷ instructions to query the database and retrieve desired results:

6.4 Capturing and Saving

6.4.1 Bill of Materials

Table 5. Bill of Materials for Capturing and Saving

Part	Cost Type	Cost	Exception
File Storage	Variable	\$0.099/GB/Month	N/A
File Write	Variable	\$0.099/100 000 Writes/Day	Free for first 20 000 writes

6.4.2 Equipment list

1. Firestore

6.4.3 Instruction

1. Setup a Firestore Database for your project. Instructions can be found [here](#)⁸.
2. Add two (2) collections: Users, Pages, (one will store). Users may be automatically created by Firebase.
3. On capture, structure your data to match a desired layout. (Example: {date: date, email: email ...})
4. Following these [instructions](#)⁹, write your structured data into Firestore.

6.5 Testing and Validation

6.5.1 Cloud Vision Tests

Table 6. Cloud Vision Test Results

Test type	Text Accuracy	Symbol Accuracy
Two page document, structured and unstructured	Medium	Very low
Single page document, structured and unstructured	High	Low
Single page document unstructured only	Very high	High
Two page document, structured and unstructured	Very low	Very low
Single page document, structured and unstructured	Low	Very low
Single page document unstructured only	Medium	Low
Two page document, structured and unstructured	Very low	Very low
Single page document, structured and unstructured	Very low	Very low
Single page document unstructured only	Moderate	Low
Single page document, structured and unstructured	Low	Very low
Single page document unstructured only	High	Moderate

* Images taken not shown for document size. All images taken from client dataset

** Scale: Very low (0%-19%), Low (20%-39%), Medium (40%-59%), Moderate (60%-79%), High (80%-89%), Very high (90%+)

From these tests, we observed that taking smaller/separate images of the structured/unstructured data provided better results. Furthermore, our symbol accuracy needs improvement and requires the OCR to be calibrated to expect symbols. Finally, another observation made is that camera quality may be a limiting factor depending on the device capturing the images since higher quality images correlated with higher accuracy scores. Structured data was not tested on this run since the feature was not implemented in this product.

6.5.2 Tesseract Tests

Due to time constraints, Tesseract OCR was not trained for matching the client's handwriting dataset. Results from the tests conducted did not provide meaningful new information.

6.5.3 Search Tests

Search tests were conducted using a simple linear search algorithm. Typesense tests were not conducted since the product is not yet implemented.

Table 7. Search Tests

Search parameters	Results
<ul style="list-style-type: none"> - Empty bar - no filters 	All pages
<ul style="list-style-type: none"> - A search term - no filters 	All pages whose content fields contain the given search term. Does not find pages with common typographical errors.
<ul style="list-style-type: none"> - Empty bar - Date Filter 	All pages whose date field match the filter. Only accepts dates in YYYY/MM/DD format
<ul style="list-style-type: none"> - Empty bar - Author Filter 	All pages whose author field exactly match the filter
<ul style="list-style-type: none"> - Empty bar - Author and Date Filter 	All pages that match both filters
<ul style="list-style-type: none"> - Search term - Author and/or Date filter 	Pages that contain the search term AND match all filters exactly.

The results concluded the validity of the search algorithm, as well as its limitations in terms of loose matches.

6.5.4 Authentication Tests

Tests were made to ensure the quality and security involved with Firebase authentication. No unregistered users were permitted access to vital website components (Search/Capture) without valid authentication. Users can only login when username and passwords exactly match as expected. Users that are already registered may not register again with the same email address.

7 Conclusion and Future Work Recommendations

There are many ways in which this product can be further developed in the future. Below are some of our recommendations for future work that could be done on this project.

7.1 Multi-Group Support

The Simply-Text app in its current iteration only supports one collection in which data is stored, accessible to all users. Implementing multi-group support would allow for the app to manage multiple collections, and for users to create new collections and assign others to that same collection. Users would be restricted to the collections they created or to the ones they were assigned to.

7.2 Improving OCR Accuracy

There exists many ways to improve the accuracy of the OCR. Outside of using training data, transforming the raw image data before extracting the text would likely improve the accuracy of the OCR. Minimizing noise near the edge of the image by cropping it, increasing the image's contrast, and sharpening the image through various algorithms could improve the readability of the image for an OCR.

Alternatively, with enough resources, training the OCR model using datasets gained from the use of the app could help in improving the OCR as well. Simply-text already stores image data along with text data, which can be used to train an OCR model in the future as long as the text is corrected by the user when first extracted.

7.3 Highlighting Detection

Storing the location and color of highlights can add another variable to search for in groups where highlights have special significance. This could be done by searching for patches of colour in an image and correlating them with word locations given by the OCR. Then, highlights can be stored as part of a document and could be searched for.

7.4 Table Detection

While this feature was planned to be implemented, it had originally relied on the assumption that the OCR was capable of detecting them automatically, which turned out to be false. Efforts were made to detect them manually, but ultimately the feature was not able to be implemented in the given time frame.

The idea behind detecting the tables manually revolved around detecting straight lines in the image, finding which boxes of text were in between which lines, and using this information to generate a 2D array of blocks of text to store. Straight line detection could be done using hough transforms after employing an image thresholding algorithm to divide the pixels into "foreground" and "background".

7.5 Conclusion

Over the course of the term, we were able to develop an app capable of capturing images, extracting text from these images, storing this text, and be able to reuse it in the future.

During the development of this app, the team faced certain obstacles which taught us some important lessons in product development and design. We learned how to search for and implement new and unfamiliar technologies at a rapid pace, how to adapt our plans when the original plan did not work as intended, and to implement redundancies to mitigate the risk of failure. The development of the app also taught us to integrate ethics into our decision-making, even at the cost of limiting our options at times. Overall, this project was an excellent exercise in creativity, project planning, and product development.

If there were a few more months to work on this project, a few things that we abandoned over the length of this process are the functions that are details to further improve the software's ability. Due to the amount of time we have, we are able to secure the basis of the client's needs and build a final prototype that suits to take on those problems provided by them. However, there are things that we abandoned midway due to other priorities, leaving us less time to deal with such components. These can include but are not limited to: testing new platforms for coding simulation and comparing them, improving the camera latency, camera focus and customization. These few things are important to add as they can improve the product that we have upfront right now.

8 Appendix I: Design Files

Table 8. Design Documents

Name	Document Location or URL	Issuance Date
Figma Mockup	Figma ¹⁰	14/10/2022
Github	Github ¹¹	29/11/2022

Table 9. Hyperlinks

Link Number	Link Text	Full Hyperlink Address
1	documentation	https://firebase.google.com/docs/auth/web/password-auth
2	readme	https://www.npmjs.com/package/firebase?activeTab=readme#user-content-use-firebase-in-your-app
3	here	https://cloud.google.com/resource-manager/docs/creating-managing-projects
4	documentation	https://cloud.google.com/vision/docs/setup
5	here	https://firebase.google.com/docs/functions/get-started
6	here	https://cloud.google.com/vision/docs/handwriting
7	these	https://firebase.google.com/docs/firestore/query-data/queries
8	here	https://firebase.google.com/docs/firestore/quickstart
9	instructions	https://firebase.google.com/docs/firestore/quickstart#add_data
10	Figma	https://www.figma.com/file/m7cG48qG7Tdqr1mVIjPWP/Prototype?node-id=0%3A3
11	Github	https://github.com/Simply-Text/simply-text